# Indexing the Directed Acyclic Graph Hierarchy of the Read Thesaurus

Erich Schulz, Nick Smejko, Colin Price, Phil Brown,
Andrzej Glowinski, Robert Hampton, Mike O'Neil.
NHS Centre for Coding and Classification, Loughborough, United Kingdom

*The hierarchical organisation of the Read Thesaurus, a fundamental feature that supports aggregation, summarisation and data entry, has previously been represented within the code itself. As Read Version 3 uses a separate table which supports multiple parents, hierarchy-based analysis becomes more complex and requires optimisation to achieve acceptable response times. Depth-first numbering of the Version 3 hierarchy allows the descendants of any node to be represented by a series of ranges and to be analysed at speeds similar to the simpler trees of earlier versions.*

## INTRODUCTION

Earlier versions of the Read Codes represent the hierarchy explicitly within the concept's code, allowing ancestral relationships to be found by inspection - a method that may be called a *path code.* For example, in Version 2 ischaemic heart disease (IHD) is coded as **G3...** (Figure 1); all descendants of ischaemic heart disease have a code beginning with **G3**. Indexing the code field of a patient record thus optimises the retrieval of this information.

| | |
|---|---|
| G.... | Circulatory system diseases |
| G3... | Ischaemic heart disease |
| G33.. | Angina pectoris |
| G331. | Prinzmetal's angina |

**Figure 1 - Extract of Read Version 2 hierarchy**

Searches in Version 3 require recursive reference to the hierarchy table, 'tree walking' downwards to identify all the concepts below IHD. The resulting set may then be matched against the patient database. Alternatively, an upwards traversal may be performed for each target concept in the patient database to see if IHD is an ancestor in the hierarchy. The implementation of both of these methods is not complex, but optimisation is required to achieve acceptable execution times.

## PATH CODES

Although the path code approach can be applied to the directed acyclic graph of Version 3, two problems arise. Firstly, as path codes have an additional character added to them at each level of the hierarchy, they can become very long (17 bytes for the current release of Version 3, and likely to increase as the thesaurus expands). Secondly, each path up to the root node must be represented explicitly, either by a separate path code for each or by linking branches by means of a separate table of sub-paths. The number of separate codes rises rapidly; sub-paths are computationally expensive.

## B NUMBERING

An alternative strategy involves depth-first, pre-order numbering of the directed acyclic graph. An example is given in Figure 2. For each node it is then possible to derive a list of ranges of numbers that represent all its descendants (Table 1). A significant feature of this method is that many of the initially derived ranges are confluent, so may be merged. Additionally, the index field is fixed length (currently less than 3 bytes), and early experiments using a test database of a million records have indicated performance improvements of two orders of magnitude when compared with tree walking. However, even with range concatenation, the number of ranges will tend to increase in proportion to the number of multiple parents within the hierarchy. Also, numbering of nodes is hierarchy-dependant, and needs to be repeated every time the hierarchy changes. For implementors of Version 3 this is every three months. To maintain integrity, references within the database need to be updated at the same time. The costs of these updates, however, are small.

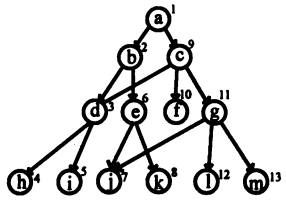


**Figure 2**

**Table 1**

| Unique identifier | Descendant ranges |
|---|---|
| a | 1-13 |
| b | 2-8 |
| c | 3-5, 7, 9-13 |
| d | 3-5 |
| e | 6-8 |
| f | 10 |
| g | 7, 11-13 |
| h | 4 |
| i | 5 |
| j | 7 |
| k | 8 |
| l | 12 |
| m | 13 |